ARTS資源予約ビデオ伝送

戸辺義人** Stephen T. Chou* 徳田英幸*
** (株) 東芝 * カーネギーメロン大学

計算機ネットワーク上でネットワーク帯域およびスレッド実行能力に応じてビデオを伝送するためのCBSRPプロトコル、CBSRPプロトコルの分散システムテストベッドARTS上へのインプリメント方法、および実験結果について述べる。ユーザの要求QOS (Quality of Service) とシステムの能力に応じて個々の伝送セション毎に、プロトコル処理のためのスレッド実行時間とFDDI同期サービス時間を予約する方法を提案する。この方法をARTSにインプリメントした結果、信頼性の高いビデオ伝送の実現可能性が示された。

Capacity-Based Video Transmission on ARTS

Yoshito Tobe** Stephen T. Chou* Hideyuki Tokuda*

*School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213, U.S.A.
**Toshiba Corp.
1, Toshiba-cho, Fuchu, Tokyo 183, Japan

To be useful in digital video applications, a computer netwo rk must support a resource allocation scheme that can guarantee a quality of service (QOS) required by its users under varying system load. We propose Capacity-Based Session Reservation Protocol (CBSRP) by which the network dynamically adjusts the allocation of Fiber Distributed Data Interface (FDDI) synchronous bandwidth and by which a thread's time for protocol processing is allocated corresponding to the remaining capacity of CPU. CBSRP has been implemented in our distributed systems testbed, Advanced Real-Time System (ARTS). The results suggest that reliable video transmission be realized by the scheme.

This research was supported in part by the U.S. Naval Ocean Systems Center under contract number N66001-87-C-0155, by the Office of Naval Research under contract number N00014-84-K-0734, by the Federal Systems Division of IBM Corporation under University Agreement YA-278067, and by the SONY Corporation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of NOSC, ONR, IBM, SONY, or the U.S. Government.

1 Introduction

Current trend in workstations is toward computing and communication which are capable of supporting video along with traditional applications [1, 4, 5, 6]. Having special-purpose hardware for video is essential but not sufficient to guarantee timely transmission digital video data. because they consume large amount of system resources and have tight timing requirements.

Video data therefore needs specialized support from the operating system through resource reservation. These resources are CPU execution time, buffers, and network bandwidth. Availability of these system resources is crucial since otherwise timing constrains inherent to video cannot satisfied due to delays caused by shortage resources. To support the reservation of these resources, several schemes have been proposed[2, 3, 4].

The resource management, however, has to be flexible to meet the demands of video users. Some users request only high quality video, while others can settle with lower quality if the system capacity unable accommodate them. Some users allow service degradation as long as a specified and agreed upon minimum quality is guaranteed.

We propose Capacity-Based Session Reservation Protocol(CBSRP) [14, 15] to support these demands. In the CBSRP, each continuous media unidirectional communication entity is called a *session*. The user can specify the minimum and maximum quality for each quality of service(QOS) parameter. The allocation of these parameters may be changed dynamically. When a new request for creating a session comes in, if the resources are already exhausted, some current sessions may be forced to reduce their QOS parameters to their minimum quality in order to accommodate the new session. The minimum quality of a session is always guaranteed once the session is created.

In this paper, we first describe the mechanism of the CBSRP. We then discuss its implementation and performance evaluation in ARTS with a FDDI (Fiber Distributed Data Interface)[9, 10] network.

2 Protocol Support

In this section, we briefly describe the CBSRP and how a user can establish a real-time communication and specify its quality.

2.1 CBSRP

The CBSRP provides guaranteed performance by reserving buffers, processor, and network bandwidth necessary for end-to-end communication. In addition, resource management keeps resource allocation in check so that no shortage of resources will occur. The capacity to time critical activities are pre-allocated according to the user's quality of service(QOS) parameters.

A *real-time session* is a unidirectional communication path between a sender and a receiver with guaranteed performance. The sender uses an established real-time session and delivers data to a remote receiver object. Each session is distinguished by a unique session identifier, registered at both sender and receiver.

In general, messages must pass through several domains[1] before reaching the receiver: sender's protocol processing domain, network domain, and receiver's protocol processing domain. In order to complete deliveries within a bounded delay, processing and delivery at each domain must be finished within a predetermined hard deadline and the sum to less than or equal to the expected end-to-end delay. If processing within each domain is schedulable to complete before the deadline, total delay of end-to-end communication is bounded. We use the deadline monotonic model [13] based on the period of the task and the worst case execution time. If a task is schedulable under the deadline monotonic policy, its deadline or delay bound within a domain can be met.

Each session is maintained by a cooperation of resource managers which consists of a session manager(SM), a system resource manager(SRM) and a

[1] Here, we used a term "domain" for indicating a "scheduling"domain as well as a "resource allocation" domain.

network resource manager(NRM). SM create, terminate, and negotiate an session with remote host. SRM checks the availability of system resources for protocol processing. NRM keeps track of total network utilization and reserve synchronous service time of 100Mbps FDDI network [9, 10].

2.2 QOS parameters for video user

Two parameters, a period and a per-period data size, determine the quality of digital video. The per-period data size is closely related to compression ratio. A smaller per-period data size enforces a higher compression ratio which will cause the quality of the media to degrade. In this sense, per-period data size corresponds to the level of resolution of media. For a receiver and a sender of video, these parameters are as important as other characteristics such as the packet loss rate or the end-to-end delay, which are common to all communications.

The user's requested value for the period and the resolution can be quantized in a finite number of classes. We define class 0 as the lowest class, class 1 as the second lowest class, and class (n-1) as the highest class when there are n classes. The number n can be set to any value. C_{prd} and C_{res} will denote the class of the period and the class of the resolution. A higher order of C_{prd} could be assigned to high-quality video service, while a lower order C_{prd} might be assigned to monitoring system. The discrete cosine transform(DCT)[8] is a commonly used method that has been incorporated in the Joint Photographic Experts Group(JPEG) [7]. Different C_{res} will result from allocating more DCT coefficients to higher level classes. The QOS parameters which the user should pass to the system are

- •minimum C_{res}, •maximum C_{res},
- •class_size[MAX_RES],
- •minimum C_{prd}, •maximum C_{prd},
- •period[MAX_PRD],
- •Importance,
- •allowable end-to-end delay, and
- •maximum packet loss rate.

MAX_RES and MAX_PRD are respectively, the number of classes for the resolution and the period defined in the system. Class_size is the data size of each C_{res} per one period. The Importance is the order of importance among sessions and is used for deciding C_{res} and C_{prd} according to the CBSRP described in the next section.

3 Implementation on ARTS

We describe the implementation of CSBRP in ARTS and FDDI network.

3.1 ARTS/FDDI

ARTS is a distributed real-time operating system being developed at CMU[11, 12]. The objective of ARTS is to develop and verify advanced real-time computing technologies. In ARTS, the basic computational entity is represented as an object. Each object may invoke both periodic and aperiodic threads. ARTS suitably supports the QOS mentioned above and thus can incorporate resource managers.

FDDI provides two types of service: synchronous service and asynchronous service. With synchronous service, access time can be bounded while the total synchronous transmission time is equal to or less than target token rotation timer(TTRT). This is suitable for periodic transmission of continuous media. FDDI is more suitable for real-time communication since it is capable of segregate non-real-time traffic from real-time traffic on the network such that real-time traffic are invariant from amount of non-real-time activities. FDDI also suitably supports the QOS mentioned above, since it can maintain the synchronous service time allocated to all sessions.

3.1.1 Protocol and Conversion of QOS

The SM and the NRM use Real-time Transport Protocol (RTP)[11] for remote invocation. RTP provides reliable, synchronous, prioritized messages.

Because session management does not need guaranteed performance, operations are delivered with the best effort on a non-real-time channel. Non-real-time traffic is sent via FDDI asynchronous service.

Continuous media data is generated periodically at very high rate. Because of short life span of the data, it is more important to keep up with its tight timing constraint rather than to ensure successful delivery or display of every arriving data. Continuous media communication needs a light protocol with high throughput rather than a reliable protocol which provides retransmission. The User Datagram Protocol(UDP) with FDDI synchronous service is thus selected with performance guaranteed by the established session through CBSRP.

The period of the user, P_{user}, and the period of media access control(MAC) are not always the same. Therefore a per-period data size in MAC D_{MAC} should be recalculated based on the MAC period P_{MAC}, P_{user}, and the user's per-period data size D_{user}. Once D_{MAC} is calculated, the user's requesting capacity D_{user}/P_{user} is measured in a one-dimensional D_{MAC}. Thus the pair of C_{res} and C_{prd} is converted to a one-dimensional class $C_{session}$. $C_{session}$ and D_{MAC} are used for dynamic QOS control in CBSRP.

3.1.2 Session Creation and Termination

Session creation may be initiated by either the sender or the receiver. Once a session is established, a periodic thread is created for the sender and a thread for receiving is created for the receiver through a user library. These threads are created for each individual session and terminated when the session ends.

To conform with CBSRP, the sender and the receiver use three library functions briefly described as follows.

```
rval=Session_Create(session_id, mpid, session, abort, relax)
rval=Session_Close(session_id)
rval=Session_Reconfig(session)
```

```
u_long *session_id;
PID *mpid; /* monitor thread pid */
struct session_dsc *session;
int *abort;
int *relax;

typedef struct session_dsc {
        u_long  sp_id;        /* session id, if known */
        OID     sp_roid;      /* remote session manager oid */
        OID     sp_serv_id;   /* sender id */
        ......
        u_long  sp_deadline;  /* suggested deadline */
        ......
        u_short max_res;      /* the maximum C_res */
        u_short min_res;      /* the minimum C_res */
        u_long  class_size[MAX_RES];
        u_short max_prd;      /* the maximum C_prd */
        u_short min_prd;      /* the minimum C_prd */
        u_long  period[MAX_PRD];
        u_short present_res;  /* present C_res */
        u_short present_prd;  /* present C_prd */
        ......
  } SM_DSC;
```

3.1.3 Resource Managers

SM, SRM, and NRM are realized as kernel objects. A single SM and SRM exist per host to handle all the session requests. The communication between a sender, a receiver, an SM, and the NRM are shown in Figure 3. Besides these messages, there are messages between an SM and an SRM checking and reserve of buffers and CPU execution time, **Buffer_Check** and **Schedulability_Check**.

The functions of SM are described in pseudo code in Figure 1. The message received by **Accept** is processed according to the type of the message.

The scheme of network bandwidth control in the NRM are shown in Figure 2 as a pseudo code. In the figure, BW[class], Max_class, Min_class, and Importance represent the network bandwidth corresponding to $C_{session}$ class, the maximum $C_{session}$, the minimum $C_{session}$, and the Importance of the requesting session, which are passed from an SM. Used_BW is the synchronous bandwidth consumed by all sessions and max_BW is the maximum synchronous bandwidth. Excess_BW[s_id]

```
while(1) {
  if(Accept(object, message, &param1) < 0) continue;
  else {
  switch(message){
   case Session_Create:
        Calculate C_session and its D_MAC;
        Buffer_Check;
        Schedulability_Check;
        Request(remote_SM, Session_Connect, &param);
        break;
   case Session_Close:
        Dealloc_Resource;
        Request(remote_SM, Session_Abort, &param);
        break;
   case Session_Connect:
        Buffer_Check;
        Schedulability_Check;
        Request(NRM, Network_Add, &param);
        if(there is any session whose class should be reduced)
           Request(sender, Network_Change, &param);
         Include acquired C_session in returning values;
         break;
   case Session_Abort:
        Dealloc_Resource;
        break;
   case Session_Reconfig:
        Calculate C_session and its D_MAC;
        Buffer_Check;
        Schedulability_Check;
        Request(remote_SM, Session_Recalc, &param);
        break;
   case Session_Recalc:
        Buffer_Check;
        Schedulability_Check;
        Request(sender, Capacity_Change, &param);
        break;
   case Network_Change:
        Request(sender, Capacity_Change, &param);
        break;
  }
  Reply(object, &rval);
 }
}
```

Figure 1: Functions of SM

```
while(1) {
if(Accept(object, message, &param1) < 0) continue;
else {
 switch(message){
   case Network_Add:
     class = Max_class;
      while( class >= Min_class ){
       if( BW[class] + Used_BW < Max_BW ){
         Used_BW += BW[class];
         Reply(object, &class);
       }
       else class--;
      }
      Left_Capacity = Max_BW - Used_BW;
      s_num = 0;
      while( Left_Capacity < BW[Min_class] ){
       s_id[s_num] = Pop_Session_Excess;
       if( Importance[s_id[s_num]] > Importance ){
        Reply(object, &FAIL);
       }
       Left_Capacity += Excess_BW[s_id[s_num]];
       s_num++;
      }
      Used_BW += BW[Min_class] - Left_Capacity;
      for( i=0; i<s_num; i++){
       if( object[s_id] <> object ){
        Request( host[s_id], Network_Change, &param);
       }
       else{
        Include s_id in rval;}
      }
      Reply(object, &rval);
      break;
   case Network_Release:
      Used_BW -= BW[class];
      Reply(object, &rval);
      break;
  }
 }
}
```

Figure 2: Bandwidth Control in NRM

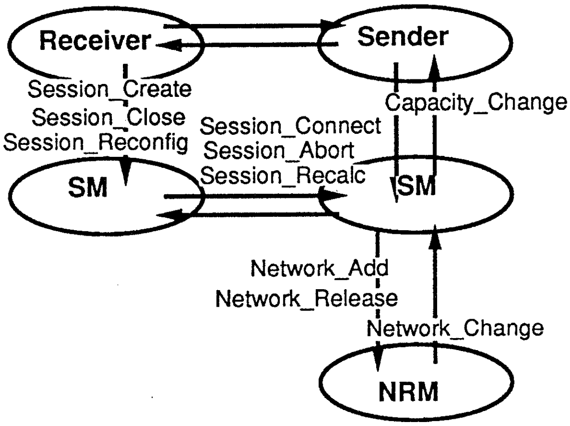

Figure 3: Communications

is the difference between BW[present $C_{session}$] and BW[minimum $C_{session}$] of a session s_id. **Pop_Session_Excess** function returns a session ID from a linked list which maintains the information on sessions in excess classes. The linked list is queued in ascending order of Importance. The return value of the **Network_Add** message is decided by considering the excess bandwidth used in the current sessions in excess classes. The **Network_Change** message is sent by the NRM to the sending SM whose session must be forced to reduce its $C_{session}$.

4 Performance Evaluation

In this section we assume that there is no overhead in capturing and displaying video so that the effect of protocol processing will be evident. We compare two cases with and without the NRM. The inter-frame times of arriving frames are measured using three SONY NEWS-1720 workstations(25 MHz MC68030) with an FDDI-adapter board SONY IKX-378 (AMD SUPERNET Chip set) and a timer board. Each workstation send or receive periodic data, or generate a background traffic. The timer consists of several counter TTL ICs with an accurate clock of 1μs granularity on the timer board.

To simulate a traffic of continuous media, two sessions, S_1 and S_2, are created. In each session, the period is fixed and only C_{res} is controlled by resource managers in order to simply verify the functions of the managers. The requesting parameters of these sessions are

maximum C_{res} = 2, minimum $Cres$ = 0,
Class_Size[0] = 8KBytes,
Class_Size[1] = 16KBytes,
Class_Size[2] = 24KBytes,
maximum C_{prd} = minimum C_{prd} = 0,
Period[0] = 30msec, and
Importance = 10.

To determine the effect of network bandwidth allocation, four cases were compared.

- *Case 1* No background traffic
- *Case 2* With background traffic 1 without NRM
- *Case 3* With background traffic 1 with NRM
- *Case 4* With background traffic 2 with NRM

The requesting parameters of the background traffic 1 are

maximum C_{res} = 0, minimum C_{res} = 0,
Class_Size[0] = 104KBytes,
maximum C_{prd} = minimum C_{prd} = 0,
Period[0] = 10msec, and
Importance = 5,

while the requesting parameters of the background traffic 1 are

maximum C_{res} = 1, minimum C_{res} = 0,
class_size[0] = 64KBytes,
class_size[1] = 104KBytes,
maximum C_{prd} = minimum C_{prd} = 0,
Period = 10msec, and
Importance = 12.

A background traffic is generated by the periodic invocation of frames with a non-existing destination address. The dummy frames are generated before the traffic initiated to avoid any additional overhead of protocol processing. For these four cases, the arriving times of FDDI frames were measured at the receiving host. The background traffic started before other sessions are created.

We set the requesting parameters to values which will clearly show the difference of each case. TTRT value is set to 30msec and Max_BW is set to 27.5msec which corresponds to 343750 bytes(= $100Mbps/8(bits/byte) * 27.5msec$). The values of Importance are chosen to compare the case in which Importance of S_1 or S_2 is higher than that of the background traffic, and the opposite case. Since D_{umax} is set to 4096 bytes, D_{MAC} of S_1 or S_2 at C_{res} = 0, 1, and 2 are 8304 bytes, 166608 bytes and 24912 bytes, respectively. D_{MAC} of the background traffic 1 and the background traffic 2 at C_{res} = 1 are 323856 bytes and 199296 bytes, respectively.

Time 0 is the time the first frame is received at the receiving host. In Figure 4 through Figure 6, ○ indicates the arriving of a 4K-byte S_1 frame and □ indicates the arriving of a 4K-byte S_2 frame. S_1 data and S_2 data are correctly delivered every 30msec in Case 1. However in Case 2, due to the heavy traffic, in excess of the network's capacity, the data are no more delivered every 30msec and the delay increases. This is validated by $323856 + 2 \times 24912 > 343750$. When the NRM is working, Case 3, both S_1 and S_2 are reduced in class, transmission of both being able to continue every 30msec. The selection of classes are ensured by $323856 + 2 \times 8304 < 343750$ and $323856 + 8304 + 166608 > 343750$. In Case 4, NRM suppresses the class of background traffic so that S_1 and S_2 are able to continue transmission at the maximum class level. The following condition is also ensured; $199296 + 2 \times 24912 < 343750$.

It is clear that NRM is avoiding the fatal error of either a long accumulated delay or a suspension of transmission. Moreover, as it is shown in Case 4, NRM enables the sessions with higher Importance to acquire the network bandwidth of other sessions.

5 Conclusions and Future Work

We demonstrated that the combination of class-based QOS parameters and dynamic allocation scheme of network bandwidth that supports such QOS parameters could facilitate flexible and predictable video transmission on ARTS with an FDDI network. We also found that the CBSRP was effective to realize the scheme. In this paper, the dynamic change was mentioned only in terms of network bandwidth. But it is also extensible to other resources such as CPU execution time.

Dynamic change of QOS with a video compression board and a further investigation on the effect of delay to QOS of video communications remain for our future study.

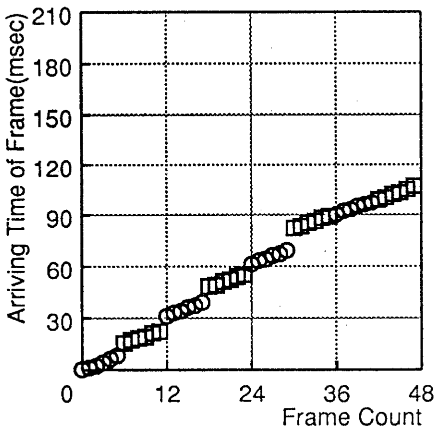

Figure 4: Timing of receiving frames for Case 1

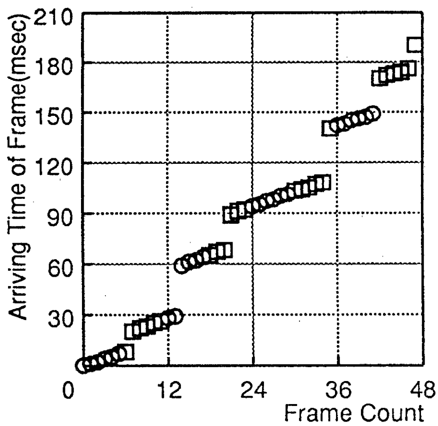

Figure 5: Timing of receiving frames for Case 2

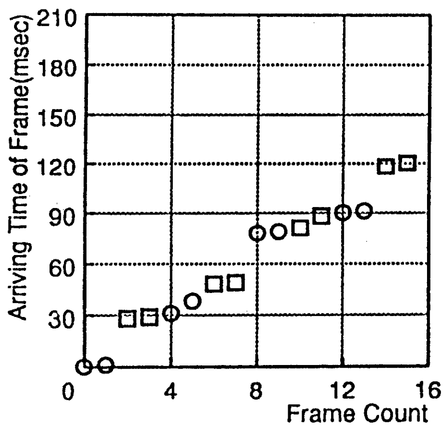

Figure 6: Timing of receiving frames for Case 3

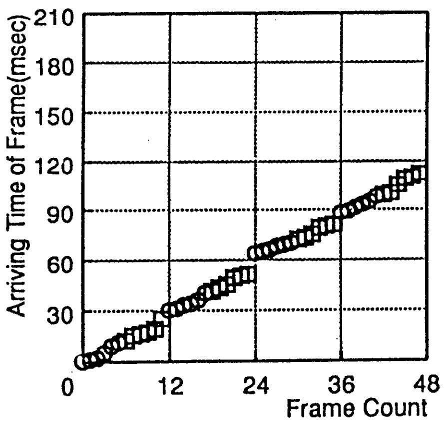

Figure 7: Timing of receiving frames for Case 4

References

[1] P. G. Milazzo, *"Shared Video under UNIX," Proc. USENIX - Summer '91*, pp.369-389.

[2] D. Ferrari and D. C. Verma, *"A Scheme For Real-Time Channel Establishment in Wide-Area Networks," IEEE J.Select. Areas Commun.*, vol.8, no.3, pp.368-379, April 1990.

[3] D. Ferrari, *"Client Requirement For Real-Time Communication Services," Network Working Group, Request for Comments:1193* Nov.1990.

[4] M. A. Rodrigues and V. R. Saksena, *"Support For Continuous Media In The DASH System," Technical Report, University of California, Berkeley*, Oct. 16, 1989.

[5] D. P. Anderson and G. Homsy, *"A Continuous Media I/O Server and its Synchronization Mechanism," IEEE Computer*, pp.51-57, Oct., 1991.

[6] H. M. Vin, P. T. Zellweger, D. C. Swinehar, and P. V. Rangan, *"Multimedia Conferencing in the Etherphone Environment," IEEE Computer*, pp.69-79, Oct., 1991.

[7] C. K. Wallace, *"The JPEG Still Picture Compression Standard," Comm. of the ACM*, vol.34, no.4, pp.30-45, Apr.1991.

[8] R. J. Clarke, *"Transform Coding of Images,"* Academic Press, New York, NY 1985.

[9] American National Standard, FDDI Token Ring Media Access Control(MAC), ANSI X3.139-1987.

[10] American National Standard, FDDI Token Ring Physical Layer Protocol(PHY), ANSI X3.148-1988.

[11] H. Tokuda, C. W. Mercer, Y. Ishikawa, and T. E. Marchok, *"Priority Inversions in Real-Time Communication" Proceedings of 10th IEEE Real-Time Systems Symposium*, December, 1989.

[12] H. Tokuda and C. W. Mercer, *"ARTS Kernel:A Distibuted Real-Time Kernel," ACM Operating Systems Review*, 23(3), July, 1989.

[13] B. Sprunt, L. Sha, and J. P. Lehoczky, *"Aperiodic Task Scheduling for Hard-real-time Systems," Journal of Real-Time Systems*, 1. pp.27-60, 1989.

[14] H. Tokuda, S. T-C. Chou, and M. Morioka, *"System Support for Continuous Media with Capacity-Based Session Reservation,"* Submitted for publication.

[15] Y. Tobe, H. Tokuda, S. T-C. Chou, and José M. F. Moura, *"QOS Control in ARTS/FDDI Continuous Media Communications,"* Submitted for publication.